CMSC 423 Homework #2:

Sequence alignment and dynamic programming

Due: Sep. 25th at the start of class

You may discuss these problems with other students, but you **must write up your solutions independently**, without using common notes or worksheets. You must indicate at the top of your homework who you worked with. Your write up should be clear, concise, and neat. You are trying to convince a skeptical reader that your algorithms or answers are correct. Messy or hard-to-read homeworks will not be graded.

1. Let v = `TACGGGTAT` and w = `GGACGTACG`. Let match cost = -1, the mismatch penalty = $+1$ and gap penalty = $+1$.

   (a) Provide a well-labeled dynamic programming table for the **global** alignment between v and w. Show backtracking arrows. Find the optimal, lowest-cost alignment and its cost.

   (b) Provide a well-labeled dynamic programming table for the **local** alignment between v and w. Show backtracking arrows. Find the optimal alignment and its cost.

   (c) Replace the linear gap penalty with an **affine gap penalty** that costs $+20$ to open a gap, and $+1$ to extend it. Find the optimal **global** alignment between v and w. Show the dynamic programming matrices.

   (Similar to problem 6.20 in your book, but where we are minimizing instead of maximizing.)

2. The SHORTEST COMMON SUPERSEQUENCE between strings x and y is the shortest string z such that both x and y are subsequences of z. (x is a "subsequence" of z if the letters of x appear in order in z, not necessarily consecutively: `at` is a subsequence of `act`).

   By modifying the basic sequence alignment algorithm, give an efficient algorithm to find the shortest common supersequence for two sequences x and y.

3. You are given strings x and y, both of length n, and a positive integer k. We are interested in the best global alignment between x and y using a simple gap penalty. Notice that because $|x| = |y|$ the gaps must come in pairs, one in x and one in y.

   Suppose the user guarantees that an optimal global alignment between x and y will have no more than k gap pairs. *Sketch* an $O(kn + n)$ dynamic programming algorithm to compute this optimal global alignment. For this problem, you don't need to provide a full algorithm, just the main idea.

   (Hint: consider first when $k = 0$. What part of the DP matrix is relevant? What about when $k = 1$?)

4. You run an ice cream business, and you want to place some advertisements in your local newspaper. There are two kinds of ads you can run, and you've noticed that Type-C works best on cold days (by promoting the good taste of your ice cream) and Type-W works best on warm days (by mentioning how cold and refreshing your ice cream is). Depending on the weather and which ad you run, you see a certain amount of increased profit that day:

	Cold	Warm
Type-C ad	+$75	+$50
Type-W ad	+$50	+$100

   You have committed to running an ad every day. The cost of placing either a Type-C or Type-W ad is $10 per day. But the newspaper charges you a fee of $25 every time you **change** which ad you are running.

   You are given a (perfectly correct) weather prediction for the next n days. Design a dynamic programming algorithm to select which ad to run on each of the next n days to maximize your profit.

   Examples:

```
Input  = WWWCCCWCWCWCW      Input  = WCW      Input  = CWWWWWC
Output = WWWCCCWWWWWWW      Output = WWW      Output = WWWWWWW
Profit = $895               Profit = $220     Profit = $530
```